Annotating an Extension Layer of Semantic Structure for Natural Language Text

Yulan Yan
The University of Tokyo
yulan@mi.ci.i.u-tokyo.ac.jp

Yutaka Matsuo
The University of Tokyo
matsuo@biz-model.t.u-tokyo.ac.jp

Mitsuru Ishizuka
The University of Tokyo
ishizuka@i.u-tokyo.ac.jp

Toshio Yokoi
Tokyo University of Technology
yokoi@media.teu.ac.jp

Abstract

Confronting the challenges of annotating naturally occurring text into a semantically structured form to facilitate automatic information extraction, current Semantic Role Labeling (SRL) systems have been specifically examining a semantic predicate-argument structure. Based on the Concept Description Language for Natural Language (CDL.nl) which is intended to describe the concept structure of text using a set of pre-defined semantic relations, we develop a parser to add a new layer of semantic annotation of natural language sentences as an extension of SRL. The parsing task is a relation extraction process with two steps: relation detection and relation classification. We advance a hybrid approach using different methods for two steps: first, based on dependency analysis, a rule-based method is presented to detect all entity pairs between each pair for which there exists a relationship; secondly, we use a feature-based method to assign a CDL.nl relation to each detected entity pair using Support Vector Machine. We report the preliminary experimental results carried out on our manual dataset annotated with CDL.nl relations.

1. Introduction

With the dramatic increase in the amount of textual information available in digital archives and on the WWW, interest in techniques for automatically extracting information from text has been growing. Identification of information from sentences and their arrangement in a structured format to be queried and used in semantic computing applications such as web searching and information extraction [4] are expected. Recently, much attention has been devoted to Semantic Role Labeling (SRL) of natural language text with a layer of semantic annotation having a predicate-argument structure, so-called shallow semantic parsing, which is becoming an important component in NLPs of various applications[14, 7]. Currently, SRL is a well-defined task with a substantial body of work and comparative evaluation[10, 3]. Within the task of semantic role-labeling, high-performance systems have been developed using FrameNet[1] and PropBank[15] corpora, respectively, as training and testing materials.

Although Semantic Role Labeling specifically examines predicate-argument structure, towards the goal of putting the whole sentence into a semantic structural form, Yokoi et al. (2005)[17] presented a descriptive language named Concept Description Language for Natural Language (CDL.nl), which is part of the realization of spirits of the work "semantic information processing"[11]. In fact, CDL.nl defines a set of semantic relations to form the semantic structure of natural language sentences in a graphical representation. They record semantic relationships showing how each meaningful entity (nominal, verbal, adjectival, adverbial) relates semantically to another entity. It connects all meaningful entities into a unified graphical representation, not only predicate-argument related entities.

Consequently, using the CDL.nl relation set, the task of structure annotation becomes a relation-extraction process that is divisible into two steps: relation detection, which is detecting entity pairs for which each there exists a meaningful relationship; and relation classification, which is labeling of each detected entity pair with a specific relation. For CDL.nl relation extraction, the challenge we must confront is that not only the relation detection step is more difficult than a classification problem as in semantic role labeling, but also that classification of a wide variation of CDL.nl relation types is harder than that of only predicate-argument roles. In this paper, we describe a hybrid approach using two different methods for each step: first, based on dependency analysis, a rule-based method is presented for relation detection; secondly, a feature-based method is presented to assign a CDL.nl relation to each detected entity pair based

978-0-7695-3279-0/08 $25.00 © 2008 IEEE
DOI 10.1109/ICSC.2008.11

on different levels of syntactic analysis.

Our contributions can be summarized as the following.

- We develop a parser to add a new layer of semantic annotation of natural language sentences. Annotation of text with a deeper and wider semantic structure can expand the extent to which shallow semantic information can become useful in real semantic computing applications such as Information Extraction and Text Summarization.
- Our study shows an intermediate phase in the progress to semantic parsing of natural language processing from syntactic processing. It will be useful to various NLP applications such as machine translation and natural language understanding.

The remainder of this paper is organized as follows. Section 2 explains the background in the semantic role labeling domain relating to semantic roles in FrameNet, PropBank, and semantic role labeling tasks. Section 3 introduces the CDL.nl relation set and specifies its importance and challenges. Section 4 proposes our hybrid method for relation extraction. Section 5 reports our preliminary experimental results and our observations. We conclude our work in Section 6.

2 Background

During the last few years, corpora with semantic role annotation and automatic annotation systems have received much attention. Three corpora are available for developing and testing predicate-argument annotation–FrameNet[1], PropBank[15], and NomBank[12]. Semantic role labeling is the process of assigning a simple WHO did WHAT to WHOM, WHEN, WHERE, WHY, HOW, etc. structure to sentences in text. In this section, we specifically address semantic role labeling systems which are based on FrameNet and PropBank.

2.1 FrameNet Semantic Roles

The Berkeley FrameNet project, started in 1998, is primarily a corpus-based lexicon-building project that documents the links between lexical items and their semantic frame(s). Its starting point is the observation that words can be grouped into semantic classes, so-called 'frames', a schematic representation of situations involving various participants, props, and other conceptual roles. Each frame has a set of predicates (nouns, verbs, or adjectives), which introduce the frame. For each semantic frame, it defines a set of semantic roles called **frame elements**, which are shared by all predicates of the frame. The term 'lexical unit' is used for a word in combination with one of its senses.

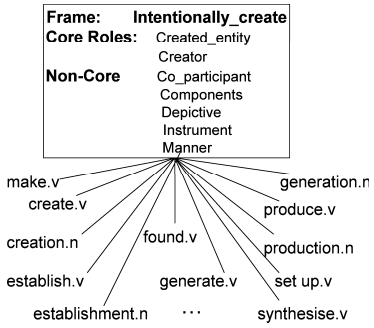

Figure 1. Sample frame from FrameNet lexicon

For example, the frame **Intentionally_create** shown in Fig. 1 is denominated using a set of semantically related predicates such as **verbs** *make* and *found*, **nouns** *creation* and *generation*, and is defined as *The Creator creates a new entity, the Created_entity, possibly out of Components.* The roles defined for this frame and shared by all its lexical entries include **core roles** *Created_entity* and *Creator*, **non-core roles** *Co_participant*, *Components*, and so on.

FrameNet contains example sentences that represent all possible syntactic and semantic contexts of the lexical items taken into consideration. In addition to the corpus, two other components distinguished in FrameNet are a set of lexical entries and a frame ontology.

Semantic role labeling processing

Based on the FrameNet annotation system, given a crude sentence, the role labeling process goes through (1) identifies all predicates, (2) disambiguates the frame for each predicate, and (3) labels the roles of arguments related to the predicate based on the frame definition.

*Bill Gates is an American entrepreneur and the [*Role *chairman] of [*Jurisdiction *Microsoft], [*Created_entity *the software company] [*Creator *he] founded [*Co_participant *with Paul Allen] [*Place *in Albuquerque, New Mexico] [*time *on April 4, 1975].*

Above is an example showing how to annotate a sentence using FrameNet roles. It is apparent that it annotates only predicate-argument roles and only for predicates "chairman" and "found", not for "entrepreneur" which is not encoded in any frame.

2.2 PropBank Semantic Roles

The FrameNet labels are rather rich in information. However, they might not always be transparent for users and annotators. The Proposition Bank (PropBank) lexicon was put forward first in 2000 to facilitate annotation, and later evolved into a resource in its own right, with the intention of adding a layer of semantic annotation to the Penn English TreeBank with verb-argument structure. Therefore, the advantage of the PropBank approach is that, using neutral la-

bels, less effort is required from annotators to assign them. Furthermore, it creates the basis for development of semi-automatic annotation of role labels, which is a requirement that must be fulfilled if we want to annotate large corpora.

Because of the difficulty in defining a universal set of semantic or thematic roles covering all types of predicates, PropBank defines semantic roles on a verb-by-verb basis. PropBank is constructed following a "bottom-up" strategy: starting from various senses of a word, a frame-file is created for every verb. Such a frame-file therefore contains all possible senses of the verb plus a set of example sentences that illustrate the context in which the verb can occur. For each sense of the verb, a role set and example sentences are available. The semantic roles covered by PropBank are the following:

Numbered arguments (*A0-A5, AA*): Semantic arguments of an individual verb are numbered beginning with 0. For a particular verb, *Arg0* is generally the argument exhibiting features of a prototypical Agent whereas *Arg1* is a prototypical Patient or Theme. The meaning of each argument label depends on the usage of the verb in question.

Adjuncts (*AM-*): General arguments that any verb might take optionally. There are 13 types of adjuncts such as *AM-ADV* (general-purpose), *AM-TMP* (temporal).

Semantic role labeling processing

Based on the PropBank annotation system, given a sentence, the role- labeling process goes through (1) and identifies each verbal predicate and (2) labels its arguments.

*Bill Gates is an American entrepreneur and the chairman of Microsoft, [*ARG1 *the software company] [*ARG0 *he] [*rel *founded] [*AM-MAN *with Paul Allen] [*AM-LOC *in Albuquerque, New Mexico] [*AM-TMP *on April 4, 1975.]*

Shown above is an example portraying how to annotate a sentence using PropBank roles. It is readily apparent that PropBank specifically examines verb predicate–argument roles.

2.3 Semantic Role Labeling Tasks

Gildea and Jurafsky[6] (2002) presented the first semantic role labeling system to apply a statistical learning technique based on FrameNet data. They describe a discriminative model for determining the most probable role for a constituent given the predicate: the frame. This task has been the subject of a previous Senseval task (Automatic Semantic Role Labeling)[10] and two shared tasks on semantic role labeling in the Conference on Natural Language Learning (2004&2005)[3].

Systems contributed to the Senseval shared task were evaluated to meet the same objectives as the Gildea and Jurafsky study using the FrameNet data. In Senseval-3, two different cases of automatic labeling of semantic roles were considered. The Unrestricted Case requires systems to assign semantic roles to the test sentences for which the boundaries of each role were given and the predicates identified. The Restricted Case requires systems to (i) recognize the boundaries of semantic roles for each evaluated frame as well as to (ii) assign a label to it. Eight teams participated in the task, with a total of 20 runs for two cases. The average precision over all Unrestricted Case runs is 0.803 and the average recall is 0.757. The average precision over all Restricted Case runs is 0.595 and the average recall is 0.481, which is notably lower than the first case, underscoring the additional difficulty of identifying the frame element boundaries.

Using CoNLL-2004, 2005, a shared task evaluated SRL systems based on the PropBank corpus. Given a sentence with several target verbs marked, a semantic role labeling system develops a machine-learning system to recognize and label the arguments of each verb predicate. In all, 19 systems participated in the CoNLL-2005 shared task. They approached the task in several ways, using different learning components and labeling strategies with different types of linguistic features, providing a comparative description and results. Evaluation is performed on a collection of unseen test sentences that are marked with target verbs and which contain only predicted input annotations; the best results in the shared task almost reached F1 at 80 in the WSJ test set, and almost 78 in the combined test.

3 CDL.nl Semantic Relation Extraction Task

Yokoi et al. (2005)[17] presented Concept Description Language for Natural Language (CDL.nl), which is used to describe the semantic/concept structure of text as a core component of W3C Common Web Language[1]. Different from existing dependency parsers, which represent the grammatical dependency structure of text, it is used to describe the semantic dependency structure of plain text in graphical form. The two basic elements for describing the structure are Entity and Relation, where the element Entity is used to represent a constituent of sentences with a head word. A set of relations[2] is defined to represent the meaning of the relationships between a pair of entities. The entity which heads the relation is called the head entity; the other one is the tail entity. A lexicon named UNLKB is used to organize entities for CDL.nl according to their semantic behaviors. they are based on their participant relations. More details about the lexicon are presented in Section 4.2.3.

3.1 CDL.nl Semantic Relation Set

With similar objectives to those of PropBank to add a layer of semantic annotation on natural language sentences,

[1] http://www.w3.org/2005/Incubator/cwl/

[2] http://www.miv.t.u-tokyo.ac.jp/mem/yyan/CDLnl/

but different from roles in PropBank, where role semantics depends on the verb and verb usage, or verb sense in a sentence, CDL.nl predefines a set of neural semantic relations covering different types of predicates. Furthermore, additional information for distinguishing similar relations is also described. For example, the definition of *aoj* (nominal entity with attribute) contains two parts:
Definition: *aoj* indicates a nominal thing that is in a state or has an attribute.
Differences between related relations: A thing with an attribute differs from *mod* in that *mod* gives some restriction of the concept that is being analyzed, whereas *aoj* signifies a thing of a state or characteristic.
Example: for the short sentence " Leaves are green", there is a relation typed as *aoj* between green and leaves, so the machine can understand that "leaves" here have the attribute "green".

Facing the challenge of defining a universal set of semantic or thematic relations covering all types of semantic relationships between entities, CDL.nl defines a set of semantic relations containing 44 relation types which are organized into three groups:

- **Intra-event relation**: Relations defining case roles, which are divided into the six abstract relations of *QuasiAgent*, *QuasiObject*, *QuasiInstrument*, *QuasiPlace*, *QuasiState*, and *QuasiTime*. Furthermore, each abstract relation includes several concrete relations which express concrete semantic information. For example, *QuasiAgent* contains five semantic relations: *agt* (agent), *aoj* (thing with attribute), *cag* (co-agent), *cao* (co-thing with attribute), *ptn* (partner). To illustrate the advantage of these subset relations, we take the *cag* (co-agent) as an example: in the sentence "John walks with Mary", "Mary" is the co-agent of event "walks". Consequently, we know both the facts that "John walks" and "Mary walks".

- **Inter-entity relations**: In addition to event-specific numbered roles, CDL.nl defines 13 more general relation types that can apply to different types of the head entity. As the definition of relation type *pur* (purpose) shows, in addition to the action entity, NominalEntity can activate the *pur* relation. Other inter-entity relations are *seq* (sequence), *equ* (equivalent), etc.

- **Qualification relations**: Relations representing qualification relationships between modified entity and modifier entity. There are nine qualification relations, collectively containing *mod* (modification), *pos* (possessor), and *qua* (quantity). This subset of relations is important to describe an entity with myriad properties.

Compared to FrameNet and PropBank, the CDL.nl relation set is useful to annotate not only facts in sentences about WHO did WHAT to WHOM or with WHOM, WHEN, WHERE, WHY, and HOW, but also What has WHICH properties, and so on. A directed graph, in which Entity is designated as a node and Relation is regarded as an arc, is useful to represent the semantic structure. An Entity is classifiable into an elemental entity and a composite entity. Composite Entity is a hyper node which contains the structure of Entity and Relation within it. It is a syntactically phrase, clause or sub-sentence. Unlike a hyper node in graphical theory, however, nodes inside and outside the Composite Entity might be mutually linked by a direct arc.

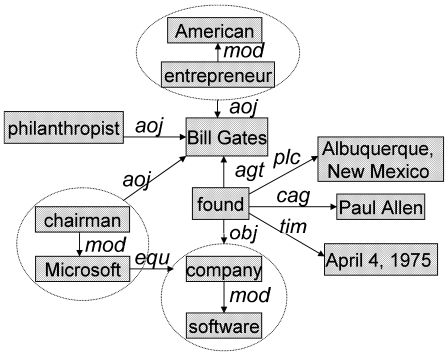

Figure 2. The graphic structure of sentence "*Bill Gates is an American entrepreneur, philanthropist and chairman of Microsoft, the software company he founded with Paul Allen in Albuquerque, New Mexico on April 4, 1975.***"**

Fig. 2 illustrates an example showing the graphical structure annotated using CDL.nl relations. Comparing to annotation with FrameNet and PropBank, it supports our idea that, with the CDL.nl relation set, plain sentences can be annotated not only using predicate-argument relations, but also that between each pair of entities, there exists a meaningful relationship, such as the *equ* (equivalent) relation between the entities "Microsoft" and "the software company", which shows that both refer to the same object, and *aoj* (thing with attribute) relation between "American entrepreneur" and "Gates", showing that "Gates" has the attribute of "American entrepreneur".

3.2 Challenges of Automatic CDL.nl Relation Extraction

The task of structure annotation with the CDL.nl relation set can be seen as a relation extraction process that is divisible into two steps: relation detection, or the detection of entity pairs between each pair for which there exists a meaningful relationship; and relation classification, or the labeling of each detected entity pair with a specific relation.

Considering the first step, semantic role detection in SRL systems involves only classifying each syntactic element in

a sentence as either a semantic argument or a non-argument by assigning a predicate, so that it is a binary-classification problem. However, the task of detecting a CDL.nl relation is not strictly a classification problem; conceptually, the system must consider all possible subsequences (i.e. consecutive words) pairs in a sentence. In this respect, the detection of dependency relations resembles that of our relation detection task. As evident from the CoNLL-X shared task on dependency parsing [2], two dominant models are currently used often for data-driven dependency parsing. The first is an "all-pairs" approach, by which every possible arc is considered in the construction of the optimal parse. The second is the "stepwise" approach, by which the optimal parse is built stepwise and where the subset of possible arcs that is considered depends on previous decisions. Clearly, the "all-pairs" approach requires exponential time in its worst case. Furthermore, although the "stepwise" approach builds a parse depending on prior decisions, our task of CDL.nl relations annotated in sentences are mutually independent. For that reason, the challenge of our first step of relation extraction is that we need an efficient method that is adequate for independent relation detection considering all possible subsequences.

For the second step, although semantic role classification involves classification of each semantic argument identified into a specific semantic role, our relation classification task involves assigning a specific CDL.nl relation to each detected entity pair to form the graphical structure of the sentence. The challenges are: 1, we must consider all 44 relation types simultaneously; 2, one major problem faced by semantic annotation of text is the fact that similar syntactic patterns might introduce different semantic interpretations and that similar meanings can be realized syntactically in many different ways.

4 Hybrid Approach for Automatic Relation Extraction

Confronting the challenges of extracting CDL.nl relations described above, in this Section, we present a hybrid approach with different methods for two steps: first, based on dependency analysis, a rule-based method is advanced for relation detection; secondly, we use a feature-based method to assign a CDL.nl relation to each detected entity pair.

4.1 Rule-based Entity Pair Identification

Language processing has been going through syntactic processing, dependency analysis, and shallow semantic parsing. To find a relationship between entities in the level of semantic processing, we use dependency analysis as the basis to perform our relation detection task because it shows the head-modifier relations between words in the level of surface-syntactic processing in a word-to-word way.

In dependency parsing[16], the task is to create links between words and name the links according to their syntactic function. By identifying the syntactic head of each word in the sentence, the analysis result is represented in a dependency graph, where the nodes are the words of the input sentence and the arcs are the binary relations from the head to dependent. Often, but not always, it is assumed that all words except one have a syntactic head, which means that the graph will be a tree with the single independent word as the root. In labeled dependency parsing, a specific type (or label) is assigned to each dependency relation that pertains between a head word and a dependent word.

Different from "all-pairs" and "stepwise" approaches, based on dependency tree structure generated from Connexor dependency parser[3], we present a rule-based method for relation detection that is done with a simple algorithm; it is depicted in Fig. 3:

- Step 1: For each input sentence, generate a dependency tree that specifies the syntactic head of each word in the sentence.
- Step 2: Find a headNode set from the dependency tree. Each can be a headword of a head entity to govern a relation. We select nodes which have subtrees and omit those which cannot be headNodes by creating a head stoplist.
- Step 3: For each headNode, check each of its subtrees to find those which can be tail entities related to the headNode. We create a tail stoplist containing those which cannot be root nodes of subtrees of tail entities. We continue to check the immediate grandchildren until reaching the leaf nodes if the root node of a subtree is in the tail stoplist.
- Step 4: A simple post-processing is applied to correct the boundaries within which the dependency tree does not show correct relationships.

As depicted in Fig. 3, for the sentence "Bill Gates found the software company with Paul Allen in Albuquerque", from the dependency tree, the following entity boundaries are generated from the dependency tree: [found, (Bill Gates)], [found, (the software company)], [found, (Paul Allen)], [found, Albuquerque] and [company, software].

4.2 Machine Learning Method for Relation Classification

When all entity pairs have been detected, facing the challenges of labeling each pair with a specific CDL.nl relation,

[3] www.connexor.com

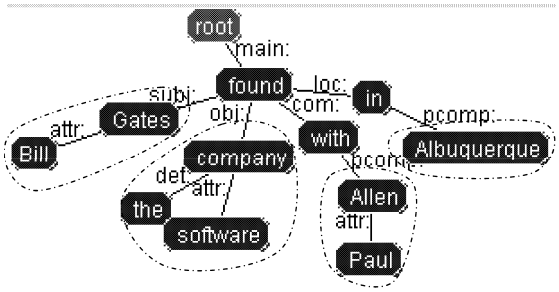

Figure 3. Relation detection example from Connexor parser

#	Text	Baseform	Syntactic relation	Syntax and morphology
1	Bill	bill	attr:>2	@A> %>N N NOM SG
2	Gates	gates	subj:>3	@SUBJ %NH N NOM SG
3	found	find	main:>0	@+FMAINV %VA V PAST
4	the	the	det:>6	@DN> %>N DET
5	software	software	attr:>6	@A> %>N N NOM SG
6	company	company	obj:>3	@OBJ %NH N NOM
7	with	with	com:>3	@ADVL %EH PREP
8	Paul	paul	attr:>9	@A> %>N N NOM SG
9	Allen	allen	pcomp:>7	@<P %NH N NOM SG
10	in	in	loc:>3	@ADVL %EH PREP
11	Albuquerque	albuquerque	pcomp:>10	@<P %NH N NOM SG
12	<s>	<s>		

Figure 4. Syntactic analysis example from Connexor

we describe a feature-based relation classification method that uses features to represent diverse knowledge of three levels of language processing: syntactic analysis, dependency parsing, and lexical construction.

4.2.1 Syntactic Features

As a benefit from the Connexor Parser, rich linguistic tags can be extracted as features to classify relations between entities. For each pair of entities of relation instances, we extract a syntactic feature set F_S containing the following features:

Morphology Features: Morphological information gives details of word forms used in the text. For example, for **Noun** words, five tags can be used: *N* (noun), *SG* (singular), *PL* (plural), *NOM* (nominative) and *GEN* (genitive). The Connexor Parser defines 70 morphology tags.

Syntax Features: Whereas morphology gives information related to forms of words, syntax describes both surface syntactic and syntactic function information of words. For example, *%NH* (nominal head) and *%>N* (determiner or premodifier of a nominal) are surface syntactic tags, *@SUB* (Subject) and *@F-SUBJ* (Formal subject) are syntactic function tags. The Connexor Parser defines 40 Syntax tags.

4.2.2 Dependency Features

For each pair of entities of relation instances, to extract a dependency feature set F_D, we define a dependency token $DT = (dep, path)$, where dep contains two labels: one is the first depend label in the dependency path, which is governed directly by the headword of head entity; the other is the final label in the dependency path pointing to the headword of participant entity. Both are closest to representing the direct dependency functions of the entity pair. In addition, $path$ is the path in the parse tree from the head entity to the other entity.

Fig. 4 portrays some examples of syntactic and dependency information of the sentence "*Bill Gates found the software company with Paul Allen in Albuquerque*". The 4th Column named syntactic relation of Fig. 4 shows dependency relations.

4.2.3 Lexical Features

To confront the problem that similar syntactic patterns might introduce different semantic interpretations, we use lexical meaning knowledge to address it in this section. Lexical meaning knowledge contains two kinds of information: word sense and semantic behavior[9].

Two lexical resources built with extensive human effort over years of work–WordNet and UNLKB–are used to capture lexical meaning knowledge. Each resource encodes a different kind of knowledge and presents its own advantages. To capture this knowledge explicitly, a set of lexical features F_L is extracted containing word-sense and word-behavior features for head words of entities:

Word-Sense Features

WordNet [5] is an on-line lexical system whose smallest unit is "synset", i.e. an equivalence class of word senses under the synonym relation. Synsets are organized by semantic relations such as Synonymy, Antonymy, and Hyponymy. In WordNet 3.0, the total of all unique nouns, verbs, adjectives, and adverb strings is actually 155287 along with 206941 word-sense pairs, containing 11529 verbs with 25047 verb-sense pairs. We use hypernymy and synonymy to represent word sense feature and also use synonymy to extend the later resource. Each word might have many hypernym senses. In our experiments, we select the top four senses.

Using the word "chairman" as an example, it has only one sense {*'chairman' in noun: president, chairman, chairwoman, chair, chairperson,*} which has eight levels of hierarchical sense, of which the top four are {{noun: living thing, animate thing}, {noun: object, physical object}, {noun: entity}, and {noun: causal agent, cause, causal

agency}}.

Word-Behavior Features

Based on the CDL.nl semantic relation set, for each usage of the word, we define semantic behavior as a series of CDL.nl semantic relations in which the word participates. Because many words have different senses and usages they might have several semantic behaviors. The UNLKB[4] is a lexicon which organizes words in a hierarchical structure according to their semantic behaviors. It includes nouns, verbs, adjectives, and adverbs and associates semantic relations in behavior representation with word type restrictions. The total of all word-behavior pairs is about 65000, containing 15000 verb-behavior pairs. It implements the close relationship between syntax and semantics for nouns, verbs, adjectives, and adverbs explicitly. Here are some word-behavior pairs of word *give* in UNLKB:

give(agt>thing,obj>thing)
give(agt>thing,gol>person,obj>thing)
give(agt>thing,gol>thing,obj>thing)
give(agt>volitional thing,obj>action)

The word *give* has semantic behaviors of at least these four kinds. Furthermore, for the second behavior, it has *agent* relation with a thing-type word, *goal* relation with a person-type word and *object* relation with a thing-type word. Here, the type of a word is a hypernym word of the word.

Because UNLKB suffers from the coverage problem, we use the synonymy set from WordNet to extend them based on the assumption: words with identical senses tend to share the same behaviors.

5 Experiments

5.1 Experimental Setting

Because this is the first work to extract CDL.nl relations from plain form text, currently no dataset exists for us to use for training and testing. After 46 person-days of discussion and manual annotation effort, we created a dataset[5] containing about 1700 sentences from Wikipedia documents. It was annotated with 13487 CDL.nl relations including 44 relation types. We evaluated the systems using ten-fold cross validation using this dataset.

To evaluate the performance of our relation classification method, we use one-vs.-all scheme in which each binary classifier will be trained for each relation label. The classifier evaluation is carried out using SVM-light software[8] with our syntactic, dependency, and lexical features.

[4] www.undl.org/unlsys/uw/unlkb.htm

[5] http://www.miv.t.u-tokyo.ac.jp/mem/yyan/CDLnl/

Table 1. Evaluation of rule-based relation detection

Task	Precision	Recall	F-value
Relation Detection	62.65	68.33	65.37

5.2 Preliminary Experimental Results

The goals of our experiments are threefold: firstly, we intend to study the performance of a rule-based relation detection method. Secondly, we evaluate our feature set for the relation classification task. Thirdly, the overall performance of relation extraction combining both steps is evaluated. For all of the purposes, three widely used evaluation measures (precision, recall and F-value) are computed.

- **Evaluation of rule-based relation detection**

For the first purpose of evaluation, the following quantities are considered to compute precision, recall, and F-value:

- p = the number of detected entity pairs.
- p+ = the number of detected entity pairs which are actual entity pairs.
- n = the number of actual entity pairs.

$$\text{Precision } (P) = \text{p+/p} \qquad \text{Recall } (R) = \text{p+/n}$$
$$F\text{-value } (F) = 2 * P * R/(P + R)$$

The results of evaluating the test file are presented in Table 1. The performance is not high. Based on error analysis of the detection results, we conclude that the reasons might be the following. 1) Some special phrases must be treated as elemental entities, whereas our algorithm generates entity pairs inside of these phrases. 2) At the level of semantic information processing, we are trying to find deeper relationships than surface function relations. In some cases, when surface analysis is not able to reflect deep semantic information directly, we must improve our detection method. 3) Some of the detection errors resulted from failures by the dependency parser.

- **Evaluation of feature-based relation classification**

For the second purpose of evaluating the performance of features for relation classification, we test three feature sets separately and the following two simple combination set, while assuming that relations have been detected correctly.

$$F_{SD} = F_S \bigcup F_D$$

Combination of syntactic and dependency features.

$$F_{SDL} = F_S \bigcup F_D \bigcup F_L$$

Combination of syntactic, dependency and lexical features.

Table 2. Preliminary performance using different features

Feature	Precision	Recall	F-value
F_S	79.33	85.78	82.43
F_D	**83.62**	**83.56**	**83.59**
F_L	73.49	81.63	77.35
F_{SD}	**85.63**	**85.91**	**85.77**
F_{SDL}	**86.35**	**87.43**	**86.89**

Table 3. Overall performance of relation extraction

TASK	Precision	Recall	F-value
Relation Detection (RD)	62.65	68.33	65.37
Relation Classification (RC)	86.35	87.43	86.89
RD + RC	51.62	57.94	54.60

Using Table 2, we can make two observations. Using different feature sets, the performance is different. This shows that each set contributes differently to our task. Another observation is that adding features continually can improve the performance, which indicates that they provide additional clues to the previous setup. Although syntax features treat two entities as independent entities; the dependency features introduce dependency connection with grammatical function information between entities. The lexical features introduce the meanings of entities, it helps in distinguishing semantic relations in case of equivalent syntactic and dependency features using word sense and usage information. By adding them into the feature vector, the performance is boosted.

- **Overall performance of relation extraction**

For the third purpose of evaluation, Table 3 presents the preliminary result of the overall performance of our relation extraction approach by combining two steps. While the performance of the relation classification step is quite adequate, the performance of relation detection is low. Despite confronting so many obstacles, CDL.nl relations were extracted using our approach with Precision, Recall, and F-values that are, respectively, 51.62, 57.94, and 54.60. Data analysis reveals that aside from dependency analysis, our method of relation detection can be improved by integrating diverse information from different levels of natural language processing.

6 Conclusions

In this paper, to surmount the challenges of semantic annotation of text, we created a new parser that (1) used a new set of semantic relations of CDL.nl, which has better coverage than those of SRL, to represent the semantic structure of text. In addition, (2) we proposed a hybrid relation extraction approach using two methods: first, based on dependency analysis, a rule-based method is presented to detect all entity pairs between each of pair for which there exists a relationship; secondly, we use a feature-based method to assign a CDL.nl relation to each detected entity pair from different levels of natural language processing. Experiments conducted using our manual dataset revealed that our approach works better to achieve relation classification than to achieve relation detection, which can be improved by integrating diverse levels of information from natural language processing.

References

[1] C.F. Baker, C.J. Fillmore, and J.B. Lowe, "The Berkeley FrameNet Project," *In Proc. COLING-ACL-98.*

[2] S. Buchholz, C.J. Fillmore, and E. Marsi, "CoNLL-X shared task on Multilingual Dependency Parsing," *In Proc. COLING-X-06.*

[3] X. Carreras and L. Marquez, "Introduction to the CoNLL-2005 shared task: Semantic role labeling," *In Proc. CoNLL-05.*

[4] P. Cimiano, M. Erdmann, and G. Ladwig, "Corpus-based Pattern Induction for a Knowledge-based Question Answering Approach," *In Proc. ICSC-07.*

[5] C. Fellbaum, WordNet: An electronic lexical database. *Cambridge, MA: MIT Press, 1998.*

[6] D. Gildea and D. Jurafsky, "Automatic Labeling of Semantic Roles," *Computational Linguistics, vol. 28, no. 3, pp. 245-288, 2002*

[7] S. Harabagiu, C.A. Bejan, and P. Morarescu, "Shallow semantics for relation extraction," *In Proc. IJCAI-05.*

[8] T. Joachims, "Text Categorization with Support Vector Machine: learning with many relevant features," *In Proc. ECML-98.*

[9] B. Levin, *English Verb Classes and Alternation: A Preliminary Investigation. The University of Chicago Press, 1993.*

[10] K. Litkowski, "Senseval-3 task automatic labeling of semantic roles," *In Senseval-3.*

[11] M. Marvin, *Semantic Information Processing. MIT Press, Cambridge, MA.*

[12] A. Meyers, R. Reeves, C. Macleod, et al. "Annotating Noun Argument Structure for NomBank," *In Proc. LREC-04.*

[13] S. Miller, H. Fox, L. Ramshaw, and R. Weischedel, *A novel use of statistical parsing to extract information from text. In 6th Applied Natural Language Processing Conference.*

[14] S. Narayanan, S. Harabagiu, "Question answering based on semantic structures," *In Proc. COLING-04.*

[15] M. Palmer, D. Gildea, and P. Kingsbury, "The proposition bank: An annotated corpus of semantic roles," *Computational Linguistics, vol. 31, no. 1.*

[16] P. Tapanainen and T. Jarvinen, *A non-projective dependency parser. In Proc. ANLP-97, Washington, D.C.*

[17] T. Yokoi, H. Yasuhara, H. Uchida, et al. *CDL (Concept Description Language): A Common Language for Semantic Computing. In WWW2005 Workshop on the Semantic Computing Initiative (SeC2005).*